Jérôme C H A I L L O U X

```
++++++++++++++++
+              +
+  O P A L     +
+              +
++++++++++++++++
```

OPAL est un programme écrit en ALGOL 510 destiné à disposer dans une surface un certain nombre de figures simples (rectangles, losanges, cercles, ellipses).

Cette surface sera assimilée à un ensemble de points ne pouvant prendre que deux états possibles (ex: blanc ou noir).

L'inclusion des figures sera obtenue par changement d'état des points à l'intérieur de ces figures.

I - Introduction des données.

En début de programme les données suivantes seront introduites sur M.A.E. :

- un nombre (NB) destiné à "lancer" la procédure aléatoire,
- le nombre de lignes (IM), de colonnes (JM) de la surface totale,
- le nombre de rectangles (NREC), de losanges (NLOS), de cercles (NCER), et d'ellipses (NELL) à y inclure.

/ PROG 012 021/

II - Calcul des paramètres.

En ALGOL 510, la déclaration d'un tableau de dimension assez grande étant impossible (par manque de place en mémoire), la surface totale ne sera pas représentée en mémoire.

Dans un premier temps, les paramètres de toutes les figures à inclure vont être calculés et stockés dans les tableaux REC, LOS, CER, ELL.

Ces paramètres sont calculés aléatoirement. Une seule règle générale est fixée : l'intersection de deux figures ne peut être
- que vide (figures disjointes),
- qu'un point,
- qu'une surface.

En aucun cas elle ne peut être un segment de droite, ceci afin de préserver la perception des figures.

Les procédures utilisées sont :
- AL, procédure aléatoire classique, pouvant relancer le programme, si les paramètres ne peuvent être trouvés.

  / PROG 100 /
- SURI, SURJ qui indiquent si les abcisses ou les ordonnées des paramètres trouvés ont déjà été utilisées.

  / PROG 200 300 /
- SURLOS qui indique si le paramètre trouvé se situe sur un périmètre de losanges. / PROG 400 /

Pour chaque rectangle on calculera deux abcisses et deux ordonnées, pour chaque losange et pour chaque cercle une abcisse, une ordonnée et un rayon, et pour chaque ellipse une abcisse, une ordonnée, un grand axe et un petit axe. / PROG 500 600 700 800 /

Les paramètres trouvés sont ensuite imprimés sur M.A.E.

III - Sortie de la figure

La surface totale est divisée en blocs de 80 colonnes sur le nombre total de lignes.

Ces blocs seront extraits un à un. La fin de chaque bloc sera signalée par l'impression sur M.A.E. du libellé FB et l'arrêt du calculateur.

/ PROG 900 /

Les blocs peuvent être

- imprimés sur M.A.E. CLE(4), sur 80 caractères, les deux états étants * et espace .

/FIG 1 /

- perforés CLE(3), sur 80 caractères avec ces mêmes états et adjonction en fin de ligne du caractère Z. La procédure standard PERF rajoutant en fin d'exécution 3 espaces et un RC, le ruban aura comme aspect en fin de ligne :
  / 80 caractères / / 3 espaces RC / / Z / / 3 espaces RC /

Ce ruban a la propriété de pouvoir être lu directement par une FLEXO (CODE CAB) déconnectée. En position "minuscules" les correspondances seront :

* (CAE)	——> ∞	espace (CAE)	——> "
RC (CAE)	——> :	Z (CAE)	——> RC

/ FIG 2 /

Problèmes posés par la taille des caractères.

Sur M.A.E CAE le rapport hauteur sur longueur d'un caractère étant voisin de 5/3 (H = 0,425 cm ; L = 0,25 cm), il en résulte une transformation de la figure.

Dans la figure 1 "l'ellipse" provient de l'équation d'un cercle et le "cercle" de l'équation d'une ellipse de rapport grand axe sur petit axe voisin de 5/3 (19/11).

Sur FLEXO CAB ce rapport est voisin de 2/1 (H = 0,42 cm ; L = 0,21 cm).

TRADUCTEUR RUBAN CAE CAB 500

Ce programme écrit en langage externe CAB permet de lire un ruban CAE. Cette lecture ne posera aucun problème car le contrôle d'imparité est le même (le nombre de perforations est impair).

D'autre part il permet de faire correspondre à deux lignes de ruban CAE une ligne sur FLEXO, ce qui conservera les dimensions exactes des figures en ramenant le rapport hauteur sur longueur des caractères voisin de 1/1.

Pour cela on utilisera les caractères
(EXP)° et espace pour la ligne haute
(IND)$_o$ et espace pour la ligne basse.

/ FIG 3/

Description du programme

1 Lecture du début du ruban jusqu'à la rencontre du caractère RC(CAB) = Z(CAE) ; ce qui fait perdre la première ligne mais évite de placer manuellement le ruban au début exact de la ligne.
/ 1,4 à 1,6 /

2 Introduction de la ligne haute en piste 2,0, de la ligne basse en piste 3,0 jusqu'à la rencontre du caractère RC(CAB)
/ 1,7 à 1,29 /

3 Comparaison du nombre de caractères introduits dans les deux lignes. Le programme s'arrête si ces deux nombres sont différents.
/ 1,30 à 1,33 /

4 Recherche de l'adresse du dernier caractère non blanc de chaque ligne. La plus grande est conservée. / 1,35 à 1,44 /

5 Impression
- pour la ligne haute (EXP)° puis espace arrière si * (CAE)
  rien si espace (CAE)
- pour la ligne basse (IND)$_o$ si * (CAE)
  espace si espace (CAE)

jusqu'à l'adresse trouvée en (4) , puis RC et retour en (2).
/ 1,57 à 1,74 /

En position "VARIANTE" allumée, le contraire se produit, créant ainsi une image négative.

OPAL 1

FB

FIG 1

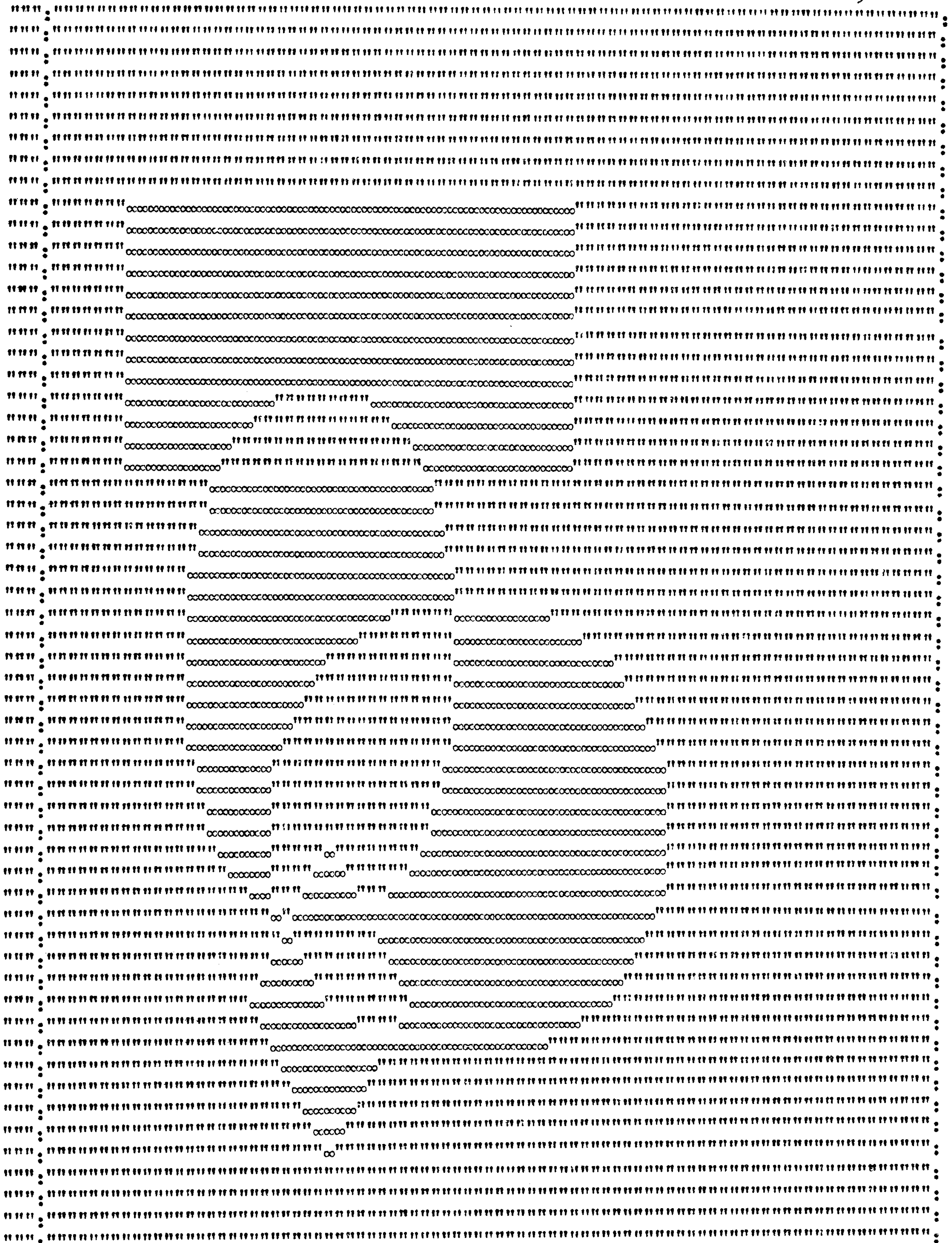


FIG -2-

1,(E

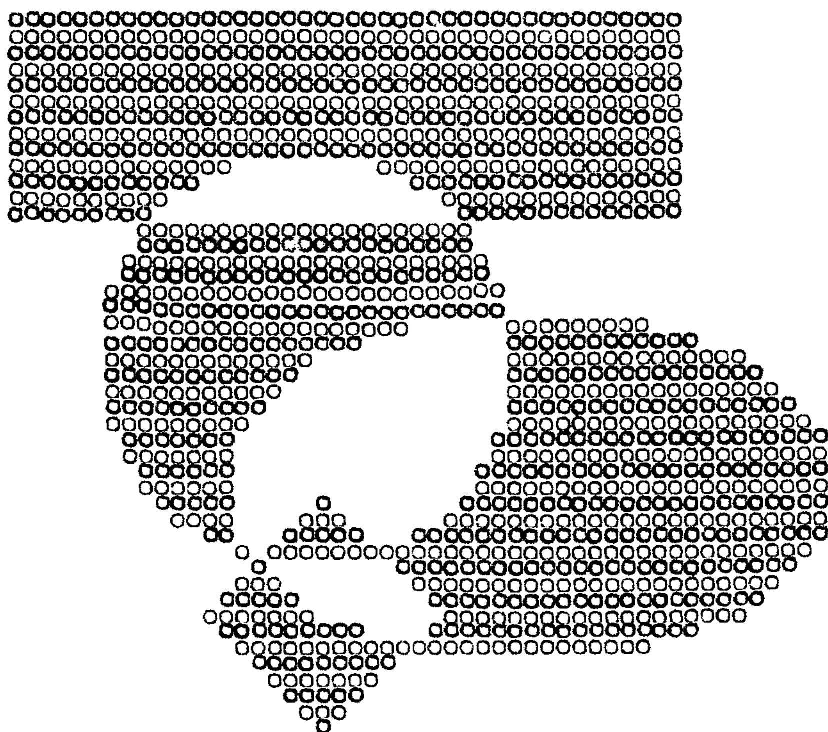


FIG -3-

CAB CØDE INTERNE	CAB HAUT BAS	RUBAN	CAE CØDE RUBAN ØCTAL	CAE HAUT BAS
00	0 E^0	-0---.----	20	+ /
01	1 E^1	00---.---0	61	
02	2 E^2	00---.--0-	62	S
03	3 E^3	-0---.--00	23	C
04	4 E^4	00---.0--	64	U
05	5 E^5	-0---.0-0	25	E
06	6 E^6	-0---.00-	26	F
07	7 E^7	00---.000	67	X
08	8 E^8	00-0.----	70	Y
09	9 E^9	-0-0.---0	31	I
10	+ .	-0-0.-0-	32	
11	- I_0	00-0.-00	73	
12	, I_7	-0-0.0--	34	
13	×)	00-0.0-0	75	(,
14	= Is	00-0.00-	76	TAB
15	ESP	-0-0.000	37	
16	/ (	000-.----	60	
17	Δ Is	-00-.---0	21	A
18		-00-.-0-	22	B
19		000-.-00	63	T
20		-00-.0--	24	D
21		000-.0-0	55	V
22		000-.00-	56	W
23	MIN	-00-.000	27	G
24	MAJ	-000.----	30	H
25	RC	0000.---0	71	Z
26	E.A.	0000.-0-	72	
27		-000.-00	33	* ;
28		0000.0--	74	
29		-000.0-0	35	
30	TAB	-000.00-	36	MIN
31	EFF.	0000.000	77	EFF.

CAB CØDE INTERNE	CAB HAUT BAS	RUBAN	CAE CØDE RUBAN ØCTAL	CAE HAUT BAS
32	AVB.	0----.----	40	- =
33	A "	-----.---0	01	1 "
34	B Is	-----.-0-	02	2 &
35	C Is	0----.-00	43	L
36	D ≤	-----.0--	04	4 >
37	E ≠	0----.0-0	45	N
38	F	0----.00-	46	Ø
39	G ?	-----.000	07	7 ±
40	H <	---0.----	10	8 _
41	I i	0--0.---0	51	R
42	J j	0--0.-0-	52	
43	K k	---0.-00	13	@
44	L ∞	0--0.0--	54	* ?
45	M ≥	---0.0-0	15	E.A.
46	N I_6	---0.00-	16	MAJ
47	Ø :	0--0.000	57	RC
48	P ≠	--0-.----	00	0 :
49	Q *	0-0-.---0	41	J
50	R ↓	0-0-.-0-	42	K
51	S £	--0-.-00	03	3 ≤
52	T >	0-0-.0--	44	M
53	U !	--0-.0-0	05	5 %
54	V I_4	--0-.00-	06	6 !
55	W I_1	0-0-.000	47	P
56	X I_2	0-00.----	50	Q
57	Y √	--00.---0	11	9 £
58	Z "	--00.-0-	12	ESP
59		0-00.-00	53	$ ≠
60		--00.0--	14	°
61		0-00.0-0	55	) :
62		0-00.00-	56	NOIR
63		--00.000	17	ROUG

CØDE - RUBAN CAB ↔ CAE

Configuration ruban

CAB	CAE
Parité	C6
C6	C5
C5	Parité
C4	C4
• Entraînement	• Entraînement
C3	C3
C2	C2
C1	C1

Le nombre de trous est impair dans les 2 cas

```
xxxxxxxxxx
x  OPAL  x
xxxxxxxxxx
```

```
001   'DEB'
002     'ENT'IM,JM,L,NB,NAL,NLOS,NREC,NCER,NELL;

100     'ENT' 'PROC' AL(N) ; 'ENT' N ;
101     'DEB' NB:=5*NB ;
102        'SI' NB 'SUG' 1093 'ALO' NB:=NB-(NB%1093)*1093 ;
103        NAL:=NAL+1 ;
104        'SI' NAL 'INF' 1000 'ALO' 'ALL' FINAL ;
105        EXL(&!#1000_AL@) ; IMPR ; NAL:=0 ;
106        'SI' CLE(7) 'ALO' 'ALL' BEGIN ;
107     FINAL:  AL:=NB-(NB%N)*N+1 ;
108     'FIN' DE LA PROCEDURE ALEATOIRE ;

010     BEGIN :
011     EXL(&!NB_IM_JM@) ; IMPR ;
012     LICLAV(NB,IM,JM) ;
013     NAL:=0;
015     L:= 'SI' IM 'SUP' JM 'ALO' JM%10 'SIN' IM%10 ;

020     EXL(&!NREC_NLOS_NCER_NELL@) ; IMPR ;
021     LICLAV(NREC,NLOS,NCER,NELL) ;

050     'DEB' 'ENT' 'TAB' LOS.(0:NLOS,1:3). , REC.(0:NREC,1:4). ,
051        CER.(0:NCER,1:3). , ELL.(0:NELL,1:4). ;
052        'BOO' 'TAB' IB.(1:IM). , JB.(1:JM). ;
053        'ENT' A,B,C,D,I,J,K,M,T ;
054        'BOO' NC,PN ;

200        'BOO' 'PROC' SURI(N) ; 'ENT' N ;
201        SURI:= IB.(N-2). 'OU' IB.(N-1).
202        'OU' IB.(N). 'OU' IB.(N+1). 'OU' IB.(N+2).;

300        'BOO' 'PROC' SURJ(N) ; 'ENT' N ;
301        SURJ:= JB.(N-2). 'OU' JB.(N-1).
302        'OU' JB.(N). 'OU' JB.(N+1). 'OU' JB.(N+2). ;

400        'BOO' 'PROC' SURLOS(N,I,J) ; 'ENT' N,I,J ;
401        'DEB' 'ENT' LA,LB,LC1,LC2,COMP ;
402           'POU' COMP:=1 'PAS' 1 'JUS' N 'FAI'
403           'DEB' LC1:=LOS.(COMP,3).+2 ; LC2:=LOS.(COMP,3).-2 ;
404              LA:=I-LOS.(COMP,1). ; LB:=J-LOS.(COMP,2). ;
405              'SI' LA+LB 'SUP' LC1 'OU' LA-LB 'SUP' LC1
406              'OU' LB-LA 'SUP' LC1 'OU' -LA-LB 'SUP' LC1
407              'ALO' 'ALL' FALSE ;
408              'SI' LA+LB 'INF' LC2 'ET' LA-LB 'INF' LC2
409              'ET' LB-LA 'INF' LC2 'ET' -LA-LB 'INF' LC2
410              'ALO' 'ALL' FALSE 'SIN' 'ALL' TRUE ;
411           FALSE : 'FIN' ;
412           SURLOS:= 'FAUX' ; 'ALL' FSURLO ;
413           TRUE : SURLOS:= 'VRAI' ;
414        FSURLO : 'FIN' DE SURLOS ;

060        'POU' K:=1 'PAS' 1 'JUS' IM 'FAI' IB.(K).:='FAUX' ;
061        'POU' K:=1 'PAS' 1 'JUS' JM 'FAI' JB.(K).:='FAUX' ;
```

```
'COM'   **  CALCUL DES PARAMETRES DES LOSANGES  ** ;
EXL(&!LOSANGES@) ; EXE(3,NLOS) ; IMPR ;
'POU' K:=1 'PAS' 1 'JUS' NLOS 'FAI'
'DEB' LOSI : C:=AL(L*5-12)+6 ;
   A:=AL(IM-2*C-6)+C+3 ;
   B:=AL(JM-2*C-6)+C+3 ;
   'SI' SURI(A+C) 'OU' SURI(A-C) 'OU' SURI(A)
   'OU' SURJ(B+C) 'OU' SURJ(B-C) 'OU' SURJ(B)
   'ALO' 'ALL' LOSI ;
   IB.(A).:=IB.(A+C).:=IB.(A-C).:=
   JB.(B).:=JB.(B+C).:=JB.(B-C).:= 'VRAI' ;
   LOS.(K,1).:=A ; LOS.(K,2).:=B ; LOS.(K,3).:=C ;
EXE(4,A,B,C) ; IMPR ; 'FIN' LOSANGES ;

'COM'   **  CALCUL DES PARAMETRES DES ELLIPSES  ** ;
EXL(&!ELLIPSES@) ; EXE(3,NELL) ; IMPR ;
'POU' K:=1 'PAS' 1 'JUS' NELL 'FAI'
'DEB' ELLI : C:=AL(IM%2-20)+10 ;
   D:=AL(JM%2-20)+10 ;
   A:=AL(IM-2*(C+5))+C+5 ;
   B:=AL(JM-2*(D+5))+D+5 ;
   'SI' SURI(A+C) 'OU' SURI(A-C) 'OU' SURJ(B+D)
   'OU' SURJ(B+D) 'ALO' 'ALL' ELLI ;
   IB.(A+C).:=IB.(A-C).:=
   JB.(B+D).:=JB.(B-D).:= 'VRAI' ;
   ELL.(K,1).:=A ; ELL.(K,2).:=B ;
   ELL.(K,3).:=C ; ELL.(K,4).:=D ;
EXE(4,A,B,C,D) ; IMPR ; 'FIN' ELLIPSES ;

'COM'   **  CALCUL DES PARAMETRES DES CERCLES  ** ;
EXL(&!CERCLES@) ; EXE(3,NCER) ; IMPR ;
'POU' K:=1 'PAS' 1 'JUS' NCER 'FAI'
'DEB' CERI : C:=AL(L*5-20)+10 ;
   A:=AL(IM-2*(C+5))+C+5 ;
   B:=AL(JM-2*(C+5))+C+5 ;
   'SI' SURI(A+C) 'OU' SURI(A-C) 'OU' SURJ(B+C)
   'OU' SURJ(B-C) 'ALO' 'ALL' CERI ;
   IB.(A+C).:=IB.(A-C).:=
   JB.(B+C).:=JB.(B-C).:= 'VRAI' ;
   CER.(K,1).:=A ; CER.(K,2).:=B ; CER.(K,3).:=C ;
EXE(4,A,B,C) ; IMPR ; 'FIN' CERCLES ;

'COM'   **  CALCUL DES PARAMETRES DES RECTANGLES  ** ;
EXL(&!RECTANGLES@) ; EXE(3,NREC) ; IMPR ;
'POU' K:=1 'PAS' 1 'JUS' NREC 'FAI'
'DEB' INDI : A:=AL(IM-8)+4 ;
   'SI' SURI(A) 'ALO' 'ALL' INDI ; IB.(A).:= 'VRAI' ;
   LONI : B:=AL(IM-8)+4 ;
   'SI' SURI(B) 'ALO' 'ALL' LONI ; IB.(B).:= 'VRAI' ;
   'SI' A 'INF' B 'ALO' 'ALL' INDJ ;
   T:=A ; A:=B ; B:=T ;
   INDJ : C:=AL(JM-8)+4 ;
   'SI' SURJ(C) 'ALO' 'ALL' INDJ ; JB.(C).:= 'VRAI' ;
   LONJ : D:=AL(JM-8)+4 ;
   'SI' SURJ(D) 'ALO' 'ALL' LONJ ; JB.(D).:= 'VRAI' ;
   'SI' C 'INF' D 'ALO' 'ALL' TSL ;
   T:=C ; C:=D ; D:=T ;
   TSL :'ALL' 'SI' SURLOS(NLOS,A,C) 'OU' SURLOS(NLOS,A,D)
   'OU' SURLOS(NLOS,B,C) 'OU' SURLOS(NLOS,B,D)
   'ALO' CHGR 'SIN' LOAD ;
   CHGR : IB.(A).:=IB.(B).:=JB.(C).:=JB.(D).:= 'FAUX' ;
   'ALL' INDI ;
   LOAD : REC.(K,1).:=A ; REC.(K,2).:=B ;
   REC.(K,3).:=C ; REC.(K,4).:=D ;
EXE(4,A,B,C,D) ; IMPR ; 'FIN' RECTANGLES ;
```

```
900     PAUSE(1) ;
901     'SI' CLE(1) 'ALO' 'ALL' BEGIN ;
902     PN:= 'SI' CLE(6) 'ALO' 'VRAI' 'SIN' 'FAUX' ;
910     'POU' M:=0 'PAS' 1 'JUS' JM%80 'FAI'
911     'DEB' EXL(&!#_OPAL_1!@) ; IMPR ;
912        'POU' I:=1 'PAS' 1 'JUS' IM 'FAI'
913        'DEB' 'POU' J:=1+M*80 'PAS' 1 'JUS' 80*(M+1) 'FAI'
914           'DEB' NC:=PN ;
920              'POU' K:=1 'PAS' 1 'JUS' NLOS 'FAI'
921              'DEB' A:=I-LOS.(K,1). ;
922                 B:=J-LOS.(K,2). ; C:=LOS.(K,3). ;
923                 'SI' A+B 'ING' C 'ET' A-B 'ING' C
924                 'ET' B-A 'ING' C 'ET' -A-B 'ING' C
925              'ALO' NC:= 'NON' NC  'FIN' ;
930              'POU'K:=1 'PAS' 1 'JUS' NREC 'FAI'
931              'SI' I 'SUG' REC.(K,1). 'ET' I 'ING' REC.(K,2).
932              'ET' J 'SUG' REC.(K,3). 'ET' J 'ING' REC.(K,4).
933              'ALO' NC:= 'NON' NC ;
940              'POU' K:=1 'PAS' 1 'JUS' NCER 'FAI'
941              'DEB' A:=I-CER.(K,1). ;
942                 B:=J-CER.(K,2). ; C:=CER.(K,3). ;
943                 'SI' A*A+B*B 'INF' C*C
944              'ALO' NC:= 'NON' NC  'FIN' ;
950              'POU' K:=1 'PAS' 1 'JUS' NELL 'FAI'
951              'DEB' C:=ELL.(K,3). ; D:=ELL.(K,4). ;
952                 A:=I-ELL.(K,1). ; B:=J-ELL.(K,2). ;
953                 'SI' A/C*A/C+B/D*B/D 'INF' 1
954              'ALO' NC:= 'NON' NC  'FIN' ;
959              'SI' NC 'ALO' EXL(&*@) 'SIN' EXL(&_@)
960           'FIN' POINT ;
961           'SI' CLE(4) 'ALO' IMPR ;
962           'SI' CLE(3) 'ALO'
963           'DEB' PERF ; EXL(&Z@) ; PERF 'FIN' ;
964           'SI' CLE(5) 'ALO'
965              'DEB' EXE(4,IM,I) ; IMPR 'FIN' ;
970        'FIN' LIGNE ;
971        EXL(&!!FB@); IMPR ;  PAUSE(5) ;
972     'FIN' BLOC ;
973     EXL(&!!FF@) ; IMPR ;
997     PAUSE(7) ;
998     'ALL' BEGIN ;
999     'FIN' 'FIN' (
```

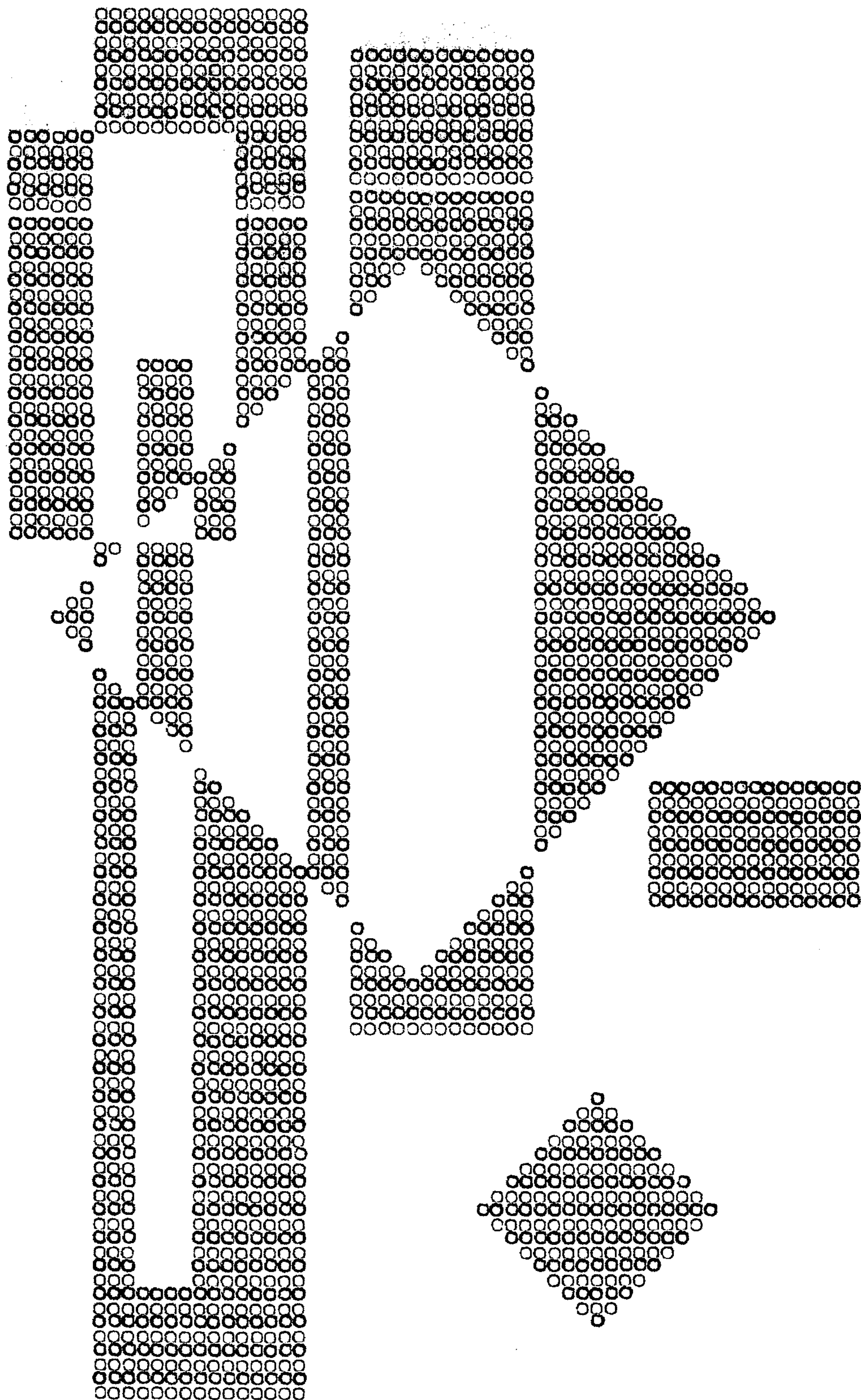
TRADUCTEUR RUBAN CAE CAB 500
****

1,0M
40PA

1,40M
40PA

1,0	EV25	1,40	V1TV44
1,1	EV25	1,41	V1TV58I
1,2	EV23	1,42	RV1,37Y13
1,3	X	1,43	A3ER15
1,4	A1EL0	1,44	A15ER2
1,5	V1TV25	1,45	X
1,6	RV1,4Y13	1,46	S15V1
1,7	A3EV1	1,47	RV1,52Z15
1,8	X	1,48	A1EM3,0+
1,9	A15EV256	1,49	V1TV44
1,10	S3V1	1,50	V1TV58I
1,11	A2EV0	1,51	RV1,46Y13
1,12	A1EL0	1,52	A4ER15
1,13	V1TV25	1,53	S15R3
1,14	S2V1Z13	1,54	A5ER4P15
1,15	V1TV58Y13	1,55	A5ER3N15
1,16	V1TV44Y13	1,56	X
1,17	RV1,12Y13	1,57	A15EV0
1,18	T1M0+	1,58	A1EM2,0+
1,19	A15V1	1,59	V1TV44
1,20	V15TV356Z3	1,60	V1TV58I
1,21	V15TV484Y3	1,61	EV0Z13
1,22	FZ13	1,62	EV26Z13
1,23	RV1,22Z13	1,63	X
1,24	X	1,64	A1EM3,0+
1,25	RV1,12Z2	1,65	V1TV44
1,26	S15V256Z3	1,66	V1TV58I
1,27	A4ER15Z3	1,67	EV11Z13
1,28	A15EV384Z3	1,68	EV15Y13
1,29	RV1,10Z3	1,69	V15TR5
1,30	S15V384	1,70	A15V1
1,31	V15TR4	1,71	RV1,58Y13
1,32	FY13	1,72	EV25
1,33	RV1,32Y13	1,73	EV23
1,34	X	1,74	RV1,7
1,35	A2ER15	1,75	X
1,36	X	1,76	X
1,37	S15V1	1,77	X
1,38	RV1,43Z15	1,78	X
1,39	A1EM2,0+	1,79	X

1 ,(E



"

